VTC-ITRI615.1

Requirements for this paper/Benodighede vir hierdie vraestel:				Resources/Hulpmiddels:
Answer Scripts/ Antwoordskrifte:		Multi-choice cards (A4)/ Multikeusekaarte (A4)		
Attendance Slips (Fill-in Paper)/ Presensiestrokies (Invulvraestel):		Graph Paper/ GrafiekPapier		
Scrap Paper/ Rofwerkpapier		Calculators/ Sakrekenaars		
Multi-choice cards (A5)/ Multikeusekaarte (A5)		Laptop (Power not provided)/ Skootrekenaar (Krag word nie voorsien nie)		

Type of Assessment/
Tipe Assessering:

Paper Number/
Vraestel Nommer:

Module Code/
Modulekode:

Examiner(s)/
Eksaminator(e):

Internal/Interne
Moderator(s):

External Moderator(s)/
Eksterne Moderator(s):

Duration/
Tydsduur:

Maximum Marks/
Maksimum Punte:

Module Description/
Module Beskrywing:

Date/
Datum:

Time/
Tyd:

Qualification/
Kwalifikasie:

Submission of answer scripts/Inhandiging van antwoordskrifte:

Instructions/Instruksies

Questions /Vrae

1. Thinking of availability (CIA), when can we refer to a data item, service or system as being available?

   As ons dink aan beskikbaarheid (CIA), wanneer kan ons verwys na 'n data item, diens of stelsel as beskikbaar? **(5)**

2. What are some of the aspects that can enhance the effectiveness of security controls?

   Wat is sommige van die aspekte wat die effektiwiteit van sekuriteits maatstawwe kan verbeter? **(4)**

3. What are some of the things a cryptanalyst can do to try and break a piece of code?

   Wat is sommige van die dinge wat 'n kriptoanalis kan doen om 'n stuk kode to probeer breek? **(6)**

4. Illustrate the working of the Merkle-Hellmann knapsack using any plaintext as a starting point.

   Illustreer die werking van die Merkle-Hellmann knapsack deur enige pleinteks as beginpunt te gebruik. **(4)**

5. Name and briefly discuss the three key principles of software engineering that assist with controlling both malicious and non-malicious program errors.

   Noem en bespreek die drie sleutel beginsels van sagteware ingenieurswese wat help met die beheer van beide kwaadwillige en nie-kwaadwillige programfoute. **(6)**

6. Discuss time-of-check to time-of-use errors.

   Bespreek tyd-van-versoek tot tyd-van-gebruik foute. **(4)**

7. If separation is the basis of protection in operating systems, what are some of the ways in which we can implement it?

As skeiding die basis van beskerming in bedryfstelsels is, wat is sommige van die maniere waarop ons dit kan implementeer? **(4)**

8. Protection of objects is a very general problem. Discuss the object access control mechanisms that exist.

   Beskerming van objekte is 'n baie algemene probleem. Bespreek die objek toegangsbeheer meganismes wat bestaan. **(7)**

9. User authentication is moving towards biometrics. What is biometrics and what are some of the problems associated with it?

   Gebruiker verifikasie is besig om te beweeg na biometrika. Wat is biometrika en wat is sommige van die probleme wat daarmee geassosieer word? **(8)**

10. When can we refer to an operating system as being trusted? Compare this to what trust is based on in trusted software.

    Wanneer kan ons verwys na 'n bedryfstelsel as vertroud? Vergelyk dit met wat vertroue op gebaseer is in

    vertroude sagteware. **(9)**

11. What are the main vulnerabilities that typical operating system flaws can be attributed to?

    Wat is die hoof kwesbaarhede waaraan tipiese bedryfstelsel foute toegeskryf kan word? **(4)**

12. Once we are aware of the vulnerabilities, we can apply assurance techniques to control them. Name and discuss the three available techniques.

    Sodra ons bewus is van die kwesbaarhede, kan ons versekeringstegnieke toepas as kontroles. Noem en bespreek die drie tegnieke. **(6)**

13. Discuss fire detection and fire suppression as it can save a lot of lives money and data.

    Bespreek vuur opsporing en onderdrukking aangesien dit baie lewens, geld en data kan spaar. **(13)**

TOTAL/TOTAAL: 80

File reference: 8.1.7.2.2